From Barnstorming to Boeing – Transforming the Internet Into a Lifeline Utility

Karl Auerbach
Chief Technical Officer – InterWorking Labs
Member Board of Directors – ICANN
http://www.cavebear.com/
http://www.iwl.com/
karl@cavebear.com

The Internet As A Utility

- The internet *is* part of our daily life.
- But we cannot fully depend on the it.
- We happen to be in an era of excess capacity
  - Things may work better today then they will in the future.
- How do we transform the net so that we can entrust our health and safety?
  - First - Improve our engineering practices.
  - Second - Rethink some of the conventional wisdom about network management.

Improving Our Engineering Practices

- Existing engineering practices are *not* going to create a utility grade internet.
- A few suggestions:
  - Testing
  - Design rules

Testing

- Test early and test often.
- Beware of testing against code with a shared genetic history.
- Demonstrations of mere interoperability are not adequate.
- Test suites and tools are a good thing.
- QA/Testing engineers deserve respect.
- Customers need effective means to report flaws.

Design Rules

- Common use in other engineering disciplines.
  - Expertise is to know when a rule is inappropriate.
- Examples:
  - Randomize timers by ±50%
  - Flexible buffer mechanisms (to avoid overrun)
  - Rollover-safe arithmetic (e.g. RFC1982)
  - Use protocol frameworks (BEEP)
- We need to be careful not to recreate the *Vasa*!

Legal Liability For Flaws

- Principle: The person or entity that creates and profits from a flawed product ought to bear the costs that that flaw causes.
  - Better product quality through fear of the consequences.
- Liability may already exist under product liability theory.
- Some circumstances may be sufficiently critical to health or safety that insurance against liability may be denied as a matter of law.
- UCITA is trying to allow vendors to repudiate responsibility for flaws.

Changing Our Engineering Conceptions

- Improving engineering practices is not sufficient to create a utility grade internet.
- Conceptual changes are needed.

Engineering Conceptions

- Fail-safe design
- Distinguish management from troubleshooting
- The net as a distributed process
- The not-so-dumb network
- Competing algorithms
- Management by delegation
- Self-healing networks
- Study network pathology
- Manageability and reparability as a primary design goals

Fail-Safe Design

- Design it so that it can not break
- Then assume that it will break.
- Then design the failure modes to be:
  - Detectable
  - Benign
  - Not contributing to larger systemic failures
- We can learn a lot from railroad and aircraft engineering.

Distinguish Network Management From Troubleshooting

- The tools and methods used for network management are fundamentally different from those used for network troubleshoting.
- Management tools can depend on the operation of fundamental network services, such as packet routing and DNS.
- Troubleshooting tools are for use when things are going south in a hurry.
  - If the net is failing to the degree that one can't keep a TCP connection open, then its time to open up the troubleshooting toolbox.

The Net As A Distributed Process

- Many manufacturers and designers view the net as a collection of independent components.
- But no device is an island.
  - Network pathologies result from the interaction of devices.
  - We can learn from process control technology.
- Network management ought to consider the management of collections of devices rather than individual devices.
  - Example: QoS routing by using ASN's as the atomic unit of management.

The Not-So-Dumb Network

- Devices *are* smart enough to participate in their own management.
  - Many vendors still do not put enough horsepower into devices to support management, fault detection, or repair.
- We need to distinguish between a dumb packet forwarding plane and a smarter control plane.

Competing Algorithms

- Mechanisms pulling in opposite ways are a good way to build stability.
  - pH stability of buffered solutions
  - Separation of powers in governments
- For every network algorithm that "does" there ought to be one that "un-does" or "does it differently"
  - Trivial cases already in use: ARP cache timeouts
- Result is a dynamic tension that hopefully leads to a usable, if not optimal, configuration.

Management by Delegation

- What to delegate:
  - Procedures
  - Resources
  - Constraints
  - Goals

Self-Healing Networks

- We've had self-healing in packet routing since the invention of packet switching.
- We need more autonomy of devices in other aspects of networking as well.
  - Example: Adaptive QoS provisioning
- Self-Healing has downsides
  - Catastrophic death spirals
  - Loss of ability to know network configuration
    - » The networking version of the Heisenberg Uncertainty Principle?

Self-Healing Networks Taking the First Steps

- Goals of Self-Healing should be narrowly defined.
  - Stick to a narrow focus
    - » E.g. QoS provisioning
  - Autonomy must be constrained
  - Autonomous acts must be tentative and must be automatically reversed if not found to improve things.
  - Healing may require human operator approval

Study Network Pathology

- We need more than anecdotal taxonomies of network failures.
- We need to get formal:
  - Network failures and errors, and their causes, need to be recorded and analyzed.
  - Cause-effect linkages need to be expressed formally.
  - Models of effect-to-cause need to be created.
- People are working on this, but it isn't sufficiently visible.

Management and Reparability From The Outset

- Management is not money wasted.
  - As the net moves to utility status, and as equipment prices drop, the cost to roll a repair truck will consume the entire profit … and more.
- Consider how automobile makers have used troubleshooting and management as a way to improve performance and add features.

Conclusion

- There is much to be done to make the net a lifeline utility.
- The work involves a convergence of many disciplines, not all of which are technical.
- This can be exciting stuff.

From Barnstorming to Boeing – Transforming the Internet Into a Lifeline Utility

Karl Auerbach
Chief Technical Officer – InterWorking Labs
Member Board of Directors – ICANN
http://www.cavebear.com/
http://www.iwl.com/
karl@cavebear.com